

ReLU as Idempotent Selection: a Row/Column View

Consider a standard feedforward ReLU network written in “gated linear” form:

$$f(x) = W_L D_{L-1} W_{L-1} \cdots D_1 W_1 x,$$

where each

$$D_k = \text{diag}(d_k) \text{ with } d_k \in \{0, 1\}^{n_k}$$

is determined by the ReLU state (which maps to 0 or 1 entry in D .)

Idempotence (why $D = D^2$ matters)

Each D_k is idempotent:

$$D_k = D_k D_k$$

This means you can freely insert or remove duplicate D_k 's without changing the function:

$$W_3 D_2 W_2 = (W_3 D_2)(D_2 W_2).$$

This “redundant factorization” is not mathematically necessary—but it is conceptually clarifying.

Two primitive operations

With D diagonal and binary:

1) Right multiplication: column selection

WD

keeps only the columns of W where $d_i = 1$, zeroing the rest.

Think: which input features are allowed through?

2) Left multiplication: row selection

DW

keeps only the rows of W where $d_i = 1$, zeroing the rest.

Think: which output features are kept?

Layer interface as “select–map–select”

Using idempotence, you can rewrite a two-layer fragment:

$W_3 D_2 W_2 D_1 W_1 x$

as

$(W_3 D_2)(D_2 W_2 D_1)(D_1 W_1)x$

Now each block has a clean interpretation:

- $D_1 W_1$: row selection $\rightarrow$ which features from layer 1 survive
- $D_2 W_2 D_1$: row + column selection $\rightarrow$ a submatrix of W_2
- $W_3 D_2$: column selection $\rightarrow$ which inputs to layer 3 are used

So every interior layer effectively applies:

$D_k W_k D_{k-1}$

i.e. a data-dependent submatrix of W_k .

Path / routing interpretation

For a fixed input x , each D_k defines an active index set:

$$I_k = \{i : (D_k)_{ii} = 1\}.$$

Then

$$D_k W_k D_{k-1} = W_k[I_k, I_{k-1}],$$

a literal submatrix.

So the network computes:

$$f(x) = W_L[I_L, I_{L-1}] \cdots W_2[I_2, I_1] W_1[I_1, :] x.$$

Interpretation: the input selects a **path of indices** through layers, and the network multiplies the corresponding chain of submatrices.

Why this helps intuition

1. Sparsity becomes explicit
You can see exactly which rows/columns (i.e., parameters) are active.
2. Separation of roles
 - D_k : *selection / routing*
 - W_k : *mixing / linear transformation*
3. Submatrix reuse
Different inputs pick different (I_{k-1}, I_k) , but all come from the same W_k .
So the model is a shared library of submatrices.
4. Local linearity
For fixed D_k 's, the whole network is one linear map built from these selected pieces.

A useful mental picture

Think of each layer as:

“Take a big matrix, punch out most rows and columns based on the data, and multiply what’s left.”

Chaining layers = multiplying a sequence of these punched-out matrices.

Subtle but important caveat

Although it’s tempting to think:

“each D independently removes ~50% of rows/columns,”

in practice:

- the selections are correlated across layers,
- earlier selections influence later ones,
- and the same weights participate in many different submatrices across inputs.

So this is not random sparsity—it’s structured, data-dependent routing.

One-line summary

Idempotent ReLU gates let you reinterpret a network as a product of data-selected submatrices, where right-multiplication selects columns, left-multiplication selects rows, and each input traces a path through a shared matrix library.